

We claim:

1. A method of providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the method comprising steps of:

- (i) receiving an incoming flow of requests from users to use an application;
- (ii) providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from respective users to use the application;
- (iii) directing each of the incoming requests to a particular instance of an appropriate resource class component;
- (iv) monitoring, for each of the users, the number of requests serviced by the instances of the resource class components of the application; and
- (v) increasing or decreasing the number of instances of one or more resource class components in response to the monitored number of requests for each resource class component.

2. The method as claimed in claim 1, further comprising the step of: directing each of the incoming requests from respective users to a particular instance of an appropriate resource class component from a respective set of one or more instances of each resource class component, said particular instance being identified as the least loaded of the instances of the appropriate resource class component from that respective set.

3. The method as claimed in claim 1, wherein the step of providing instances of each resource class component further comprises the steps of:

initiating one or more instance of one or more resource class on a plurality of machines to service incoming requests to use the application; and

terminating one or more instances of each resource class on a plurality of machines to service incoming requests to use the application.

4. The method as claimed in claim 1, wherein requests from users to use the application are stored in a queue for execution by a particular instance of the appropriate resource class on a first-in-first-out basis.

5. The method as claimed in claim 1, further comprising the step of: maintaining a record of service obligations to respective users.

6. The method as claimed in claim 5, further comprising the step of: increasing or decreasing, for each of the users, the number of instances of each resource class component in response to the monitored number of requests for each resource class component, wherein the service obligations to respective users are at least met.

7. The method as claimed in claim 1, further comprising the step of: maintaining a record of the current rate of requests received from respective users, based on the monitored number of serviced requests.

8. The method as claimed in claim 7, wherein said step of increasing or decreasing the number of instances of said one or more resource classes is (i) at least partly based upon said recorded current rate of requests received from respective users, and (ii) at least partly based on predetermined information that correlates changes in request rates with charges in the corresponding number of instances of said one or more resource classes required to service said request rates.

9. The method as claimed in claim 1, wherein one or more of the users are organizations, and the requests are generated by individuals associated with the respective organization.

10. A method of defining a charging structure for charging a user for providing access to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the method comprising the steps of:

defining an expected range of rates of requests from a user to use an application, from (i) an expected minimum rate of requests from a user, to (ii) an expected maximum rate of requests from a user;

defining servicing charges associated with servicing rates of requests for one or more subranges within the expected range; and

defining a penalty charge payable in favour of the user if: (i) one or more requests from the user are not serviced; and (ii) said one or more unserviced requests occur in a period when the rate of requests from the user did not exceed the expected maximum rate of requests.

11. The method as claimed in claim 10, wherein the servicing charges are defined as a unit cost for each request from the user.

12. The method as claimed in claim 11, wherein the unit cost varies depending on: (i) the physical machine on which the request is serviced; and/or (ii) the resource class(es) involved in servicing the request; and/or (iii) the computational complexity associated with servicing the request.

13. A method of providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the method comprising steps of:

receiving an incoming flow of requests from users to use an application;

providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from the users to use the application;

monitoring, for each of the users, the resources currently available and resources currently consumed by the requests serviced by instances of the resource class components of the application; and

maintaining (i) a record of resources currently available to respective users; and (ii) a record of resources currently consumed by respective users; both records of said resources being maintained in respect of each of the one or more instances of each resource class components.

14. The method as claimed in claim 13, further comprising the step of:

adjusting the respective numbers of said one or more instances of each resource class component;

wherein said instances of each resource class component are adjusted for each user based (i) at least partly on said records of resources currently available and currently consumed by respective users, and (ii) at least partly on predetermined information that estimates the number of each resource class components required to service requests for said instances of the resource class components.

15. A system for providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the system comprising:

(i) means for receiving an incoming flow of requests from users to use an application;

(ii) means for providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from respective users to use the application;

- (iii) means for directing each of the incoming requests to a particular instance of an appropriate resource class component;
- (iv) means for monitoring, for each of the users, the number of requests serviced by the instances of the resource class components of the application; and
- (v) means for increasing or decreasing the number of instances of one or more resource class components in response to the monitored number of requests for each resource class component.

16. A computer software program, recorded on a medium and capable of execution by computing means able to interpret the computer software program, for providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the computer software program comprising:

- (i) code means for receiving an incoming flow of requests from users to use an application;
- (ii) code means for providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from respective users to use the application;
- (iii) code means for directing each of the incoming requests to a particular instance of an appropriate resource class component;
- (iv) code means for monitoring, for each of the users, the number of requests serviced by the instances of the resource class components of the application; and

- (v) code means for increasing or decreasing the number of instances of one or more resource class components in response to the monitored number of requests for each resource class component.

17. A system for defining a charging structure for charging a user for providing access to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the system comprising:

means for defining an expected range of rates of requests from a user to use an application, from (i) an expected minimum rate of requests from a user, to (ii) an expected maximum rate of requests from a user;

means for defining servicing charges associated with servicing rates of requests for one or more subranges within the expected range; and

means for defining a penalty charge payable in favour of the user if: (i) one or more requests from the user are not serviced; and (ii) said one or more unserviced requests occur in a period when the rate of requests from the user exceeded the expected maximum rate of requests.

18. A computer software program, recorded on a memory medium and capable of execution by computing means able to interpret the computer software program, for method of defining a charging structure for charging a user for providing access to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the computer software program comprising:

code means for defining an expected range of rates of requests from a user to use an application, from (i) an expected minimum rate of requests from a user, to (ii) an expected maximum rate of requests from a user;

code means for defining servicing charges associated with servicing rates of requests for one or more subranges within the expected range; and

code means for defining a penalty charge payable in favour of the user if: (i) one or more requests from the user are not serviced; and (ii) said one or more unserved requests occur in a period when the rate of requests from the user do not exceed the expected maximum rate of requests.

19. A system for providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the system:

means for receiving an incoming flow of requests from users to use an application;

means for providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from the users to use the application;

means for monitoring, for each of the users, the resources currently available and resources currently consumed by the requests serviced by instances of the resource class components of the application; and

means for maintaining (i) a record of resources currently available to respective users; and (ii) a record of resources currently consumed by respective users; both records of said resources being maintained in respect of each of the one or more instances of each resource class components.

20. A computer software program recorded on a medium and able to be executed by computing means able to interpret the computer software program, for providing access for a plurality of users to an application comprising a plurality of resource class components collectively executing on multiple networked machines, the computer software program comprising:

code means for receiving an incoming flow of requests from users to use an application;

code means for providing, for each of the users, respective sets of one or more instances of each resource class component for the application on one or more machines, to service the incoming requests from the users to use the application;

code means for monitoring, for each of the users, the resources currently available and resources currently consumed by the requests serviced by instances of the resource class components of the application; and

code means for maintaining (i) a record of resources currently available to respective users; and (ii) a record of resources currently consumed by respective users; both records of said resources being maintained in respect of each of the one or more instances of each resource class components.

TO BE USED